UPPSALA
UNIVERSITET

Introduction

Syntactic analysis (5LN455)

2014-11-12

Sara Stymne
Department of Linguistics and Philology

Mostly based on slides from Marco Kuhlmann

Today

- Introduction to syntactic analysis
- Course information
- Exercises

What is syntax?

- Syntax addresses the question how sentences are constructed in particular languages.
- The English (and Swedish) word *syntax* comes from the Ancient Greek word *sýntaxis* ‘arrangement’.

What is syntax not?

Syntax does not answer questions about …

… how speech is articulated and perceived
(phonetics, phonology)

… how words are formed
(morphology)

… how utterances are interpreted in context
(semantics, pragmatics)

What is syntax not?

Syntax does not answer questions about …

… how speech is articulated and perceived (phonetics, phonology)

… how words are formed (morphology)

… how utterances are interpreted in context (semantics, pragmatics)

simplified

Why should you care about syntax?

- Syntax describes the distinction between well-formed and ill-formed sentences.
- Syntactic structure can serve as the basis for semantic interpretation and can be used for
  - Machine translation
  - Information extraction and retrieval
  - Question answering
  - …

Parsing

The automatic analysis of a sentence with respect to its syntactic structure.

Theoretical frameworks

- Generative syntax

  Noam Chomsky (1928–)

- Categorial syntax

  Kazimierz Ajdukiewicz (1890–1963)

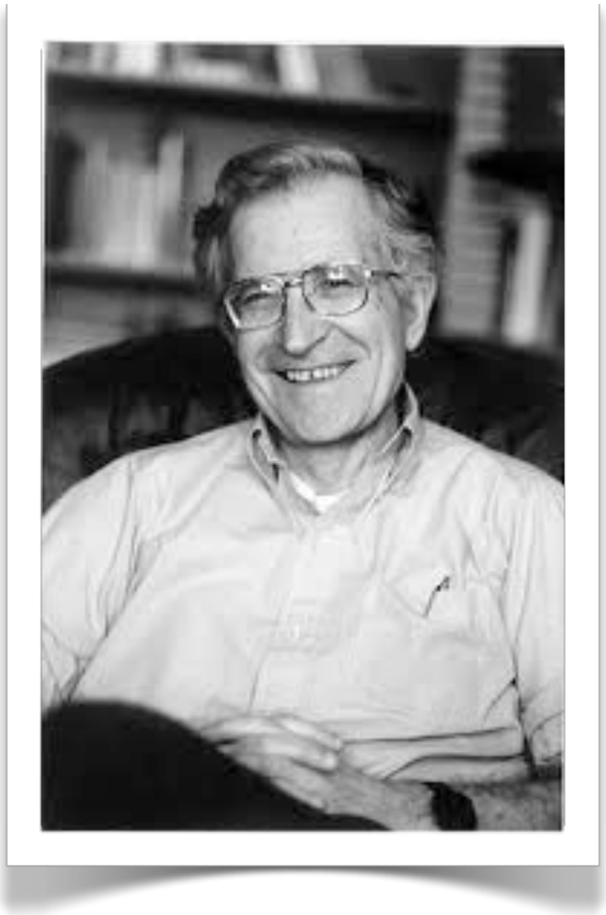
- Dependency syntax

  Lucien Tesnière (1893–1954)

Theoretical frameworks

- Generative syntax
  Noam Chomsky (1928–)
- Categorial syntax
  Kazimierz Ajdukiewicz (1890–1963)
- Dependency syntax
  Lucien Tesnière (1893–1954)

Theoretical frameworks

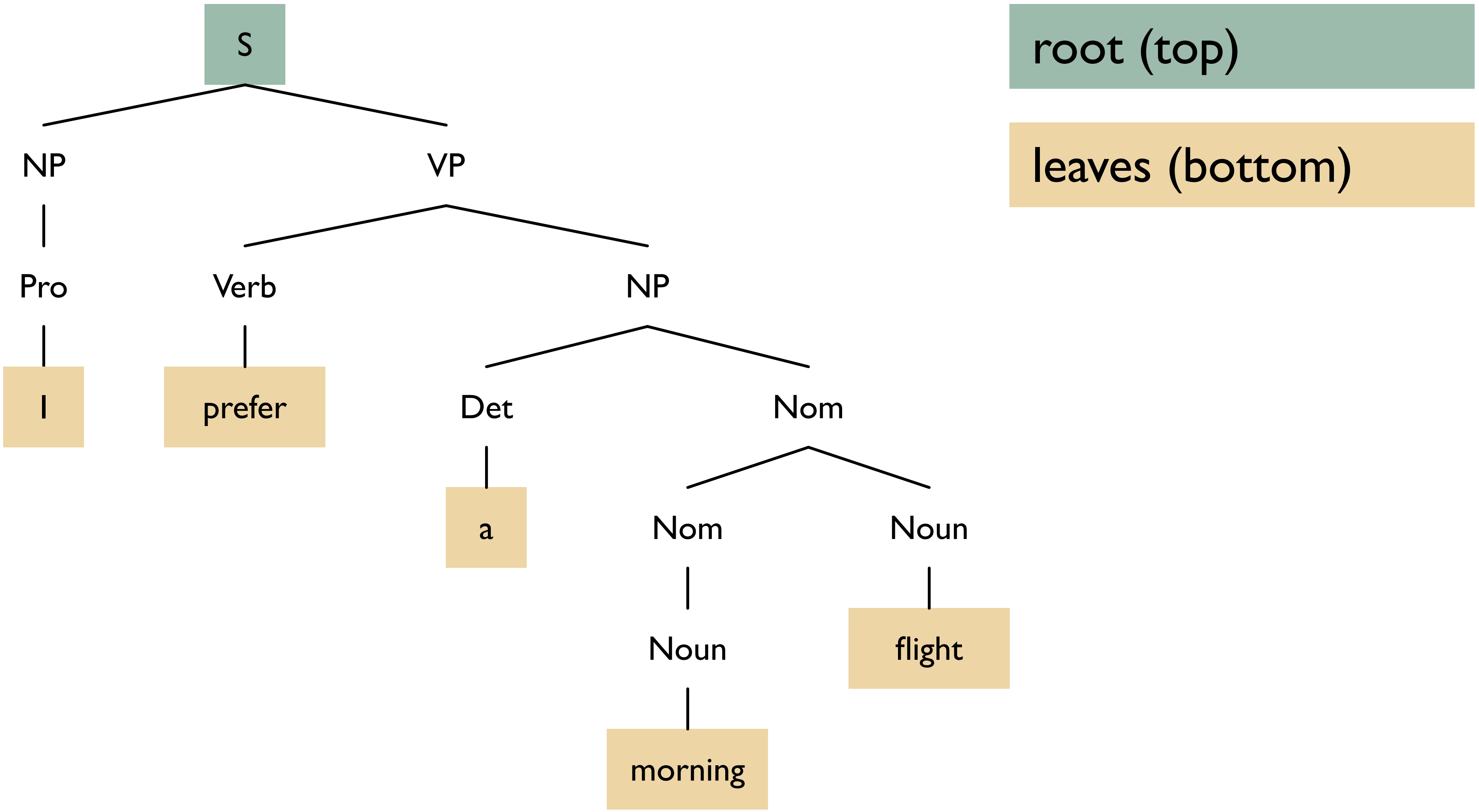


Chomsky

Ajdukiewicz

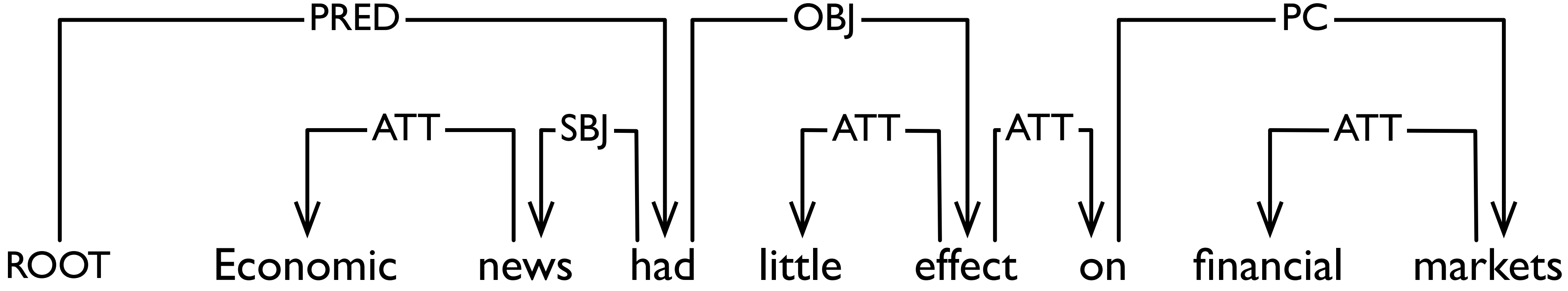
Tesnière

Phrase structure trees



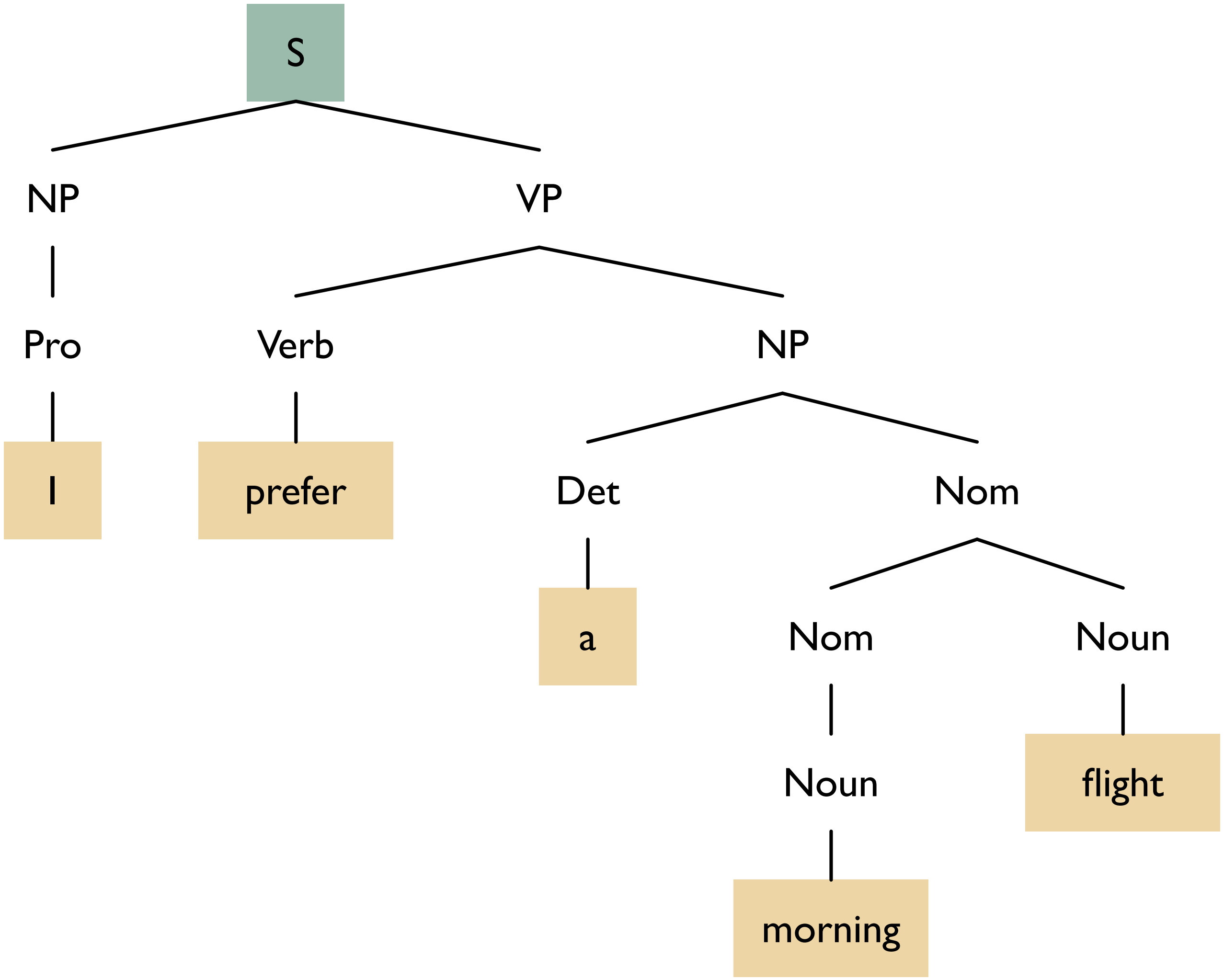
UPPSALA
UNIVERSITET

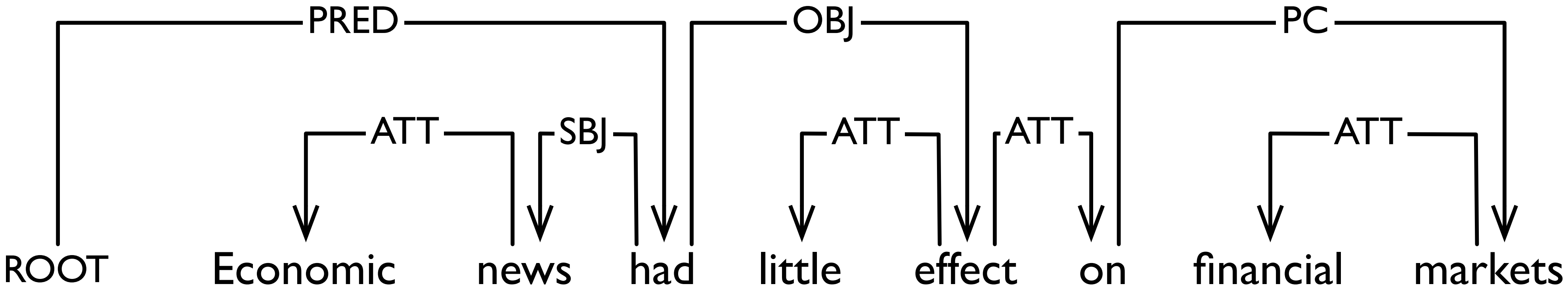
Dependency trees



UPPSALA
UNIVERSITET

Phrase structure vs dependency trees



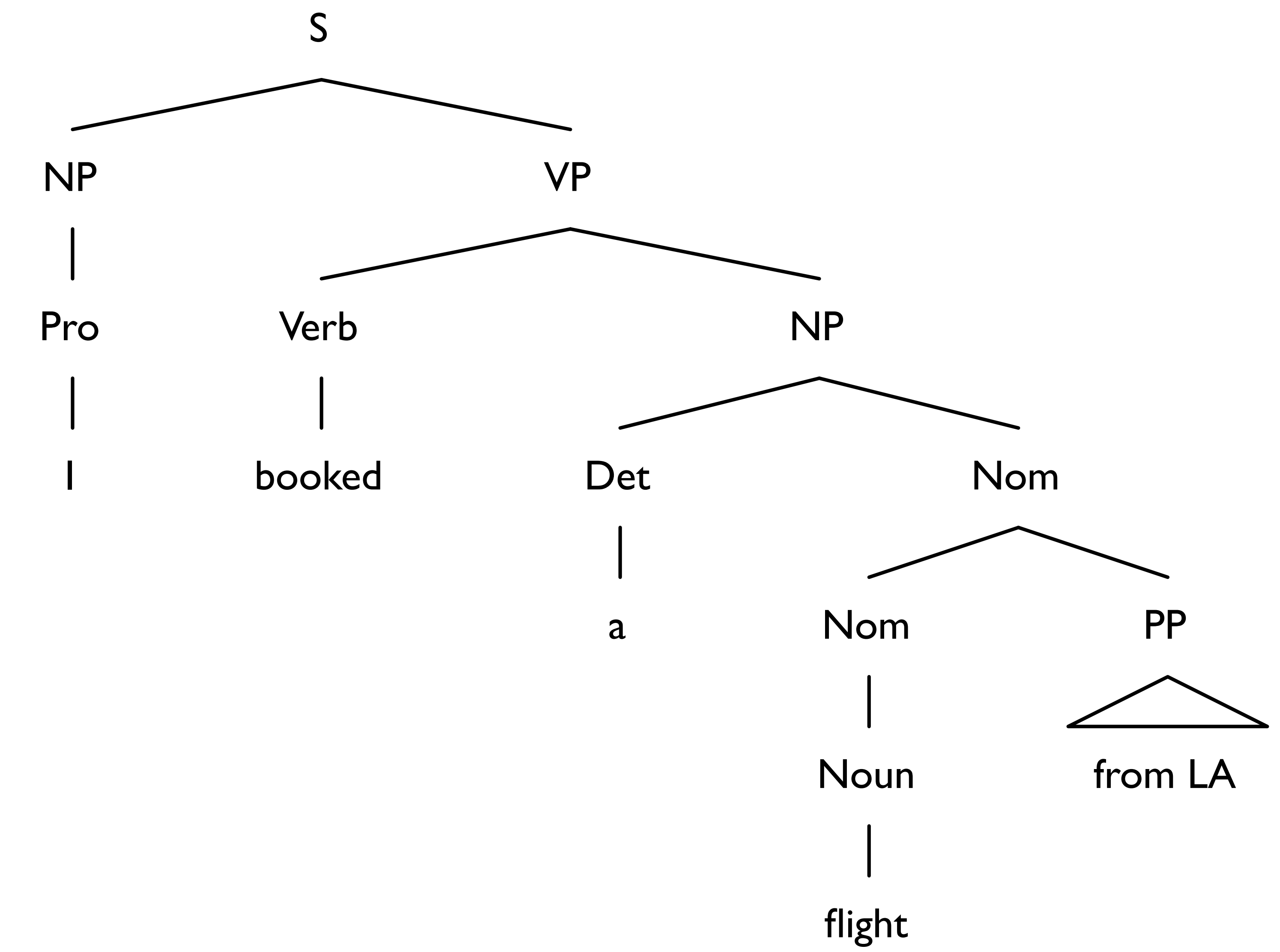


Ambiguity

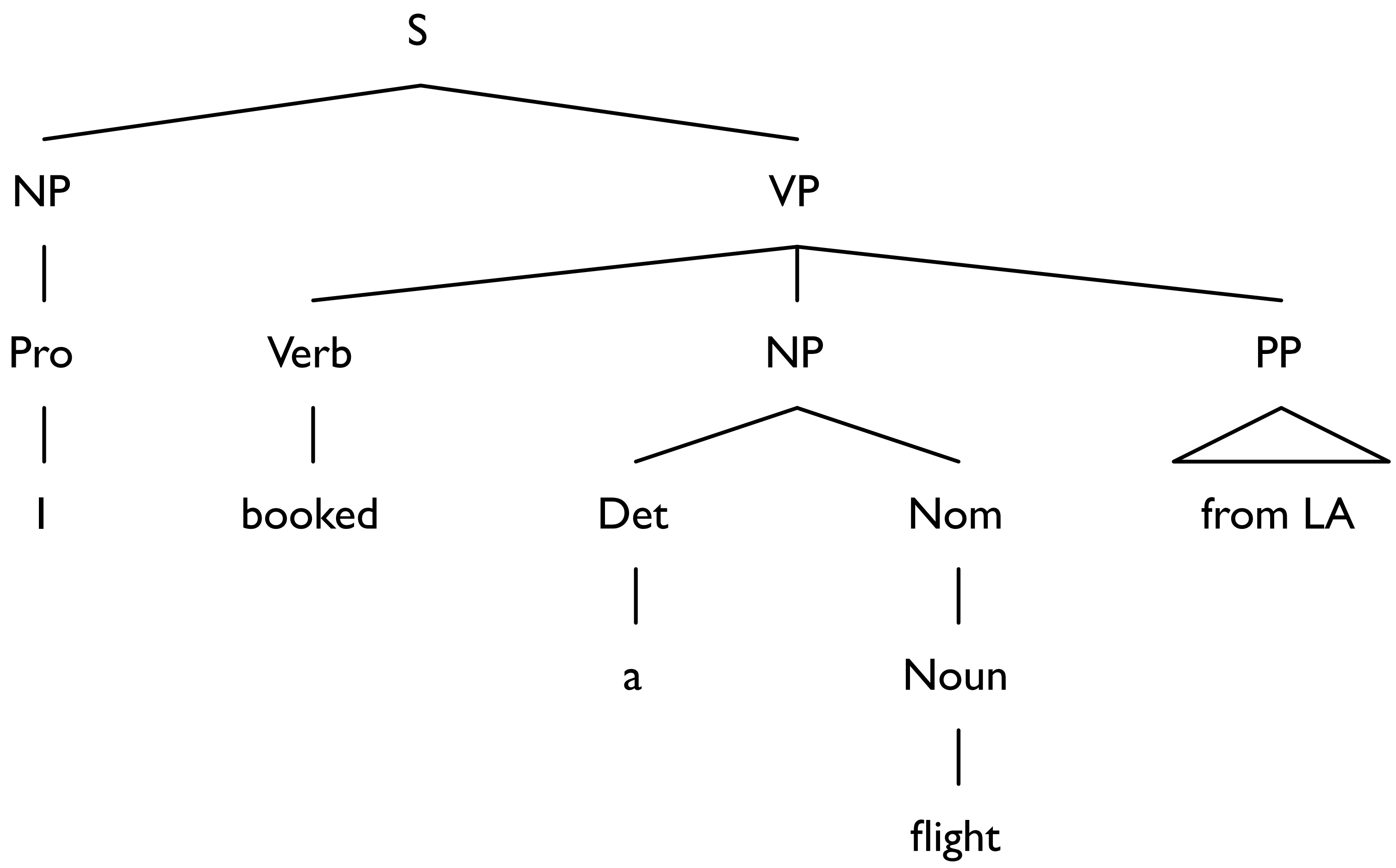
I booked a flight from LA.

- This sentence is ambiguous. In what way?
- What should happen if we parse the sentence?

Ambiguity



Ambiguity



Interesting questions

- Is there any parse tree at all?
- What is the best parse tree?

Parsing as search

- Parsing as search:
  Search through all possible parse trees
  for a given sentence.
- In order to search through all parse trees
  we have to 'build' them.

Top–down and bottom–up

top–down

only build trees that are rooted at *S*

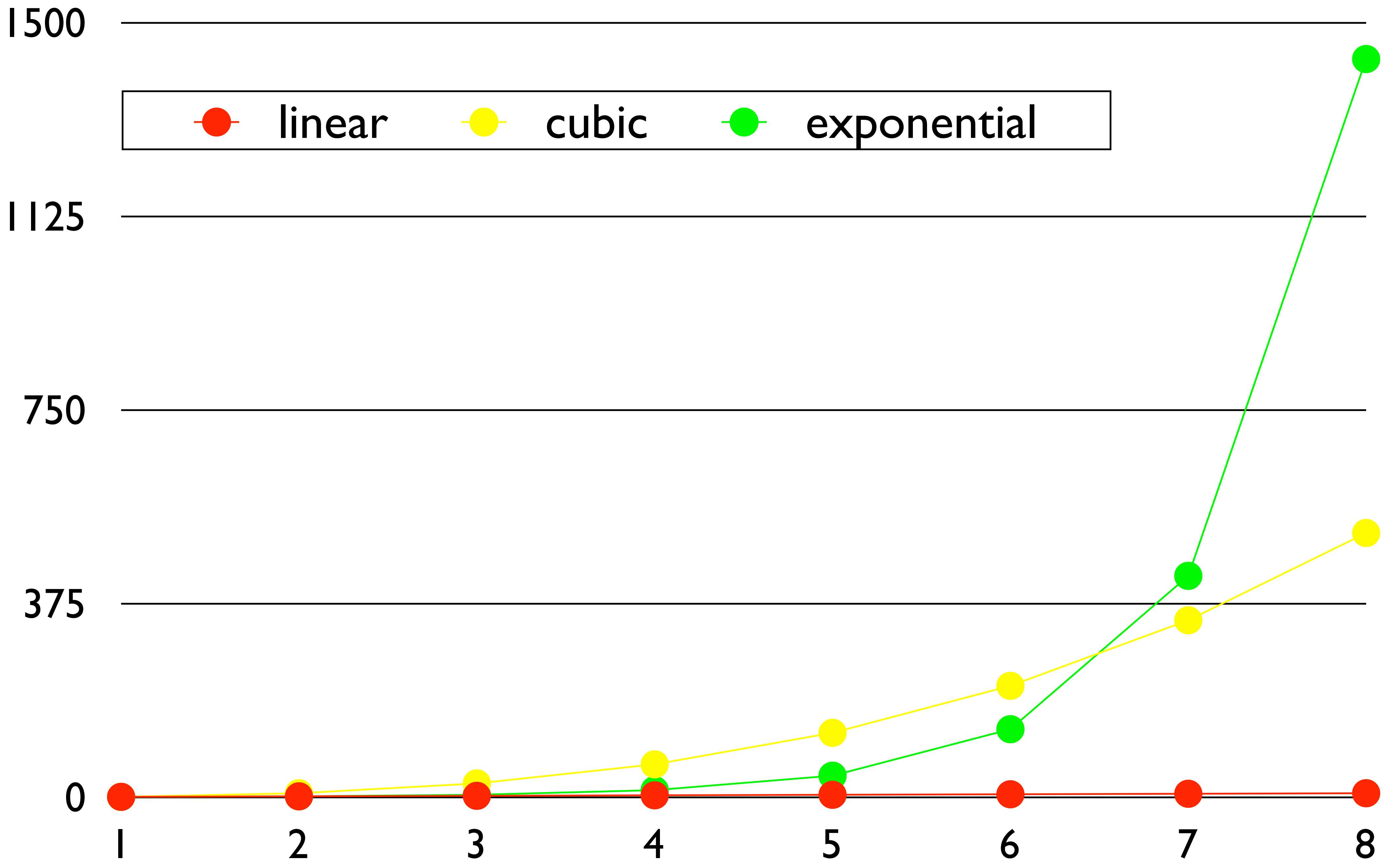
may produce trees that do not match the input

bottom–up

only build trees that match the input

may produce trees that are not rooted at *S*

How many trees are there?



Dynamic programming (DP)

- Divide and conquer:
  In order to solve a problem, split it into subproblems, solve each subproblem, and combine the solutions.
- Dynamic programming (DP):
  Solve each subproblem only once and save the solution in order to use it as a partial solution in a larger subproblem.
- Memoisation:
  Solve only the necessary subproblems and store their solutions for resue in solving other subproblems.

Complexity

- Using DP we can (sometimes) search through all parsetrees in polynomial time.
- That is much better than to spend exponential time!
- But it may still be too expensive!
  In these cases one can use an approximative method such as greedy search or beam search.

Course information

Intended learning outcomes

At the end of the course, you should be able to

- account for the parsing problem of phrase structure grammar and dependency grammar;
- explain at least two different methods for automatic syntactic analysis: one for phrase structure parsing, one for dependency parsing;
- account for statistical methods for syntactic disambiguation;

Intended learning outcomes

- apply existing systems that use these methods to realistic data and evaluate them with respect to their accuracy and efficiency;
- implement a central component of at least one approach to syntactic analysis in a suitable programming language.

Examination

- Examination is continuous and distributed over four graded assignments and two literature seminars.
- Two assignments are one-page papers.
  Time to invest: about 8 hours per assignment.
- The other two assignments are small projects where you need to implement/test parsers.
  Time to invest: about 40 hours per assignment.
- In the seminars you will discuss scientific articles about parsing.
  Time to invest: about 10 hours per seminar

Assignments

1. Written assignment on phrase structure parsing
2. Programming assignment: implement CKY parsing
3. Written assignment on dependency parsing
4. Use and evaluate an exisiting system for dependency parsing

Literature seminars

- Read one given article for each seminar
- Prepare according to the instructions on the homepage
- Everyone is expected to be able to discuss the article and the questions about it
  - It should be clear that you have read and analysed the article, but it is perfectly fine if you have misunderstood some parts
- The seminars are obligatory
  - If you miss a seminar or are unprepared, you will have to hand in a written report.

Learning outcomes and examination

- account for the parsing problem of phrase structure grammar and dependency grammar; paper assignments + seminars
- explain at least two different methods for automatic syntactic analysis: one for phrase structure parsing, one for dependency parsing; paper assignments + seminars
- account for statistical methods for syntactic disambiguation; paper assignments + seminars

Learning outcomes and examination

- apply existing systems that use these methods to realistic data and evaluate them with respect to their accuracy and efficiency; project assignment 2
- implement a central component of at least one approach to syntactic analysis in a suitable programming language. project assignment 1

Grading

- The assignments are graded with G and VG
- G on the seminars if present, prepared and active. The seminars are obligatory!
- To achieve G on the course:
  - G on all assignments and seminars
- To achieve VG on the course:
  - Same as for G **and** VG on at least two assignments, of which at least one is practical

Teachers

- Sara Stymne
  - Course coordinator, lectures, seminars, assignments
- Joakim Nivre
  - Examiner

Teaching

- 8 lectures
- 2 seminars
- No scheduled supervision / lab hours
- Supervision available on demand:
  - Email
  - Knock on office door
  - Book a meeting

Course workload

- 7.5 hp means about 200 hours work
- 16 h lectures
- 4 h seminar
- 180 h work on your own
    - ~ 96 h assignment work
    - ~ 20 h seminar preparation
    - ~ 64 h additional reading

Schedule

week 46 (2 hrs taught, 14 hrs reading, 4 hrs assignments)		
12/11	13–15	Lecture: Introduction
week 47 (4 hrs taught, 6 hrs reading, 10 hrs assignments)		
17/11	10–12	Lecture: Phrase structure parsing 1
19/11	10–12	Lecture: Phrase structure parsing 2
week 48 (4 hrs taught, 4 hrs seminar prep, 12 hrs assignments)		
24/11	13–15	Lecture: Phrase structure parsing 3
26/11	13–15	Lecture: Phrase structure parsing 4
week 49 (2 h taught, 6 hrs seminar prep, 12 hrs assignments)		
03/12	10–12	Literature seminar 1 (maybe in half class, times might change slightly)

Schedule

week 50 (2 hrs taught, 8 hrs reading, 10 hrs assignments)		
10/12		Deadline: Assignments 1 and 2
10/12	15–17	Lecture: Dependency parsing 1
week 51 (4 hrs taught, 12 hrs reading, 4 hrs assignments)		
15/12	10–12	Lecture: Dependency parsing 2
17/12	10–12	Lecture: Dependency parsing 3
week 52-02 (0 hrs taught, 4 hrs seminar prep, 24 hrs reading, 32 hrs assignments)		
week 03 (2 hrs taught, 6 hrs seminar prep, 14 hrs assignments)		
14/01	13–15	Literature seminar 2 (maybe in half class, times might change slightly)
16/01		Deadline: Assignments 3 and 4

Reading: course books

- Daniel Jurafsky and James H. Martin.
  Speech and Language Processing. 2nd edition.
  Pearson Education, 2009.
  Chapters 12-14.
- Sandra Kübler, Ryan McDonald,
  and Joakim Nivre. Dependency Parsing.
  Morgan and Claypool, 2009.
  Chapter 1-4, 6.

Reading: articles

- Seminar 1
  - Mark Johnson. PCFG Models of Linguistic Tree Representations. Computational Linguistics 24(4). Pages 613-632.
- Seminar 2
  - Joakim Nivre and Jens Nilsson. Pseudo-Projective Dependency Parsing. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). Pages 99-106. Ann Arbor, USA.

Evaluation from last year

- Overall score: 4.75
  - Scores above 4 in all categories
  - Overall students found the course interesting, and the assignments good
- This year
  - Two literature seminars instead of one general seminar
  - Small changes to assignment 4
  - Otherwise not much change, since the course was working well

Work to do this week

- Read chapter 12.1-12.7
- Read chapter 13.1-13.3 in preparation for next Monday
- Read descriptions of assignment 1 and 2
- If you need, repeat parts of grammar course from last year, related to phrase structure grammars and dependency grammars